เรื่องน่าอ่าน
สมชาย นำประเสริฐชัย

สารพัน UNIX

ตอน เชลล์...ล่ามผู้ปราดเปรื่อง (2)

ศึกษาเพิ่มเติมเรื่องของเชลล์ซึ่งเป็นส่วนหนึ่งที่ทำให้ระบบยูนิกซ์มีประสิทธิภาพสูงขึ้น

จากตอนที่แล้วในฉบับที่ 83 ได้กล่าวถึงความเป็นมาและรูปแบบของเชลล์อย่างคร่าว ๆ และในครั้งนี้ก็เป็นการกล่าวถึงเชลล์เพิ่มเติม และโครงสร้างเงื่อนไขต่าง ๆ ที่ทำให้คอร์นเชลล์มีประสิทธิภาพเทียบได้กับภาษาชั้นสูงทีเดียว รวมถึงการสร้างบรรยากาศในการทำงาน

สัญลักษณ์พิเศษในเชลล์สคริปต์มีด้วยกันอยู่ 4 ชนิดคือ

1. Double quote (") เป็นการกำหนดค่าตัวแปรโดยมีช่องว่างระหว่างคำได้

2. Single quote (') จะไม่ทำการตีความหมายตัวแปร แต่จะให้ค่าออกมาโดยตรงเลย

3. Back Slash character (\) มีความหมายเหมือนกับ [2] แต่การใช้จะยุ่งยากกว่า

4. Single back quote (`) เป็นสัญลักษณ์ที่ใช้เป็นตัวกำหนดค่าตัวแปรคำสั่ง และในคอร์นเชลล์รุ่นหลังวันที่ 6 มี.ค. 86 สามารถใช้สัญลักษณ์ $ (...) แทนได้ ดังตัวอย่างรูปที่ 1

ตัวดำเนินการต่าง ๆ ในเชลล์สคริปต์

ส่วนหนึ่งที่ทำให้ขีดความสามารถของคอร์นเชลล์ เทียบได้กับภาษาชั้นสูงก็คือ ที่มีการใช้ตัวดำเนินการในด้านต่าง ๆ ทั้งทางคณิตศาสตร์, ตรรกศาสตร์ อีกทั้งยังมีตัวดำเนินการเกี่ยวกับไฟล์อีกด้วย

ตัวดำเนินการเงื่อนไข

คำสั่ง 1; คำสั่ง 2 คำสั่งที่ 2 จะเริ่มทำงานก็ต่อเมื่อการทำงานในคำสั่งที่ 1 สิ้นสุดแล้ว

คำสั่ง 1 & คำสั่ง 2 คำสั่งที่ 1 และ 2 จะเริ่มทำงานไปพร้อม ๆ กัน

คำสั่ง 1 && คำสั่ง 2 คำสั่งที่ 2 จะทำงานก็ต่อเมื่อคำสั่งที่ 1 เป็นจริงคือให้ค่าเป็น 0

คำสั่ง 1 : : คำสั่ง 2 คำสั่งที่ 2 จะทำงานก็ต่อเมื่อคำสั่งที่ 1 เป็นเท็จคือให้ค่าที่ไม่เป็น 0

รูปที่ 1 การใช้สัญลักษณ์แทน

```
$ word="Hello UNIX World"
$ echo $word
  Hello UNIX World
$ echo "$word"
  Hello UNIX World
$ echo '$word'
  $word
$ echo '!@#55&& '
  !@#55&&
$ echo \$word
  $word
$ echo \C \O \M \P \U \T \E \R
  C O M P U T E R
$ echo "date"
  date
$ echo `date`
  Thu Apr 09 15:57:40 THT 1992
$ echo $(date)
  Thu Apr 09 15:57:48 THT 1992
```

ตัวดำเนินการทางคณิตศาสตร์

ค่าที่ 1 –eq ค่าที่ 2 ค่าตัวเลขที่ 1 เท่ากับค่าตัวเลขที่ 2

ค่าที่ 1 –ne ค่าที่ 2 ค่าตัวเลขที่ 1 ไม่เท่ากับค่าตัวเลขที่ 2

ค่าที่ 1 –lt ค่าที่ 2 ค่าตัวเลขที่ 1 น้อยกว่าค่าตัวเลขที่ 2

ค่าที่ 1 –le ค่าที่ 2 ค่าตัวเลขที่ 1 น้อยกว่าหรือเท่ากับค่าตัวเลขที่ 2

ค่าที่ 1 –gt ค่าที่ 2 ค่าตัวเลขที่ 1 มากกว่าค่าตัวเลขที่ 2

ค่าที่ 1 -ge ค่าที่ 2 ค่าตัวเลขที่ 1 มากกว่าหรือเท่ากับค่าตัวเลขที่ 2

ตัวดำเนินการที่เกี่ยวกับไฟล์

ตรวจสอบสถานะของไฟล์

–f เป็นไฟล์ธรรมดา

–d เป็นไดเรคทอรี

–c เป็นไฟล์พิเศษ (ประเภท character)

–b เป็นไฟล์พิเศษ (ประเภท block)

–r เป็นไฟล์ที่อนุญาตให้อ่านได้

–w เป็นไฟล์ที่อนุญาตให้เขียนได้

–x เป็นไฟล์ที่อนุญาตให้เอ็กซีคิวต์ได้

–s เป็นไฟล์ที่มีขนาดมากกว่า 0

เปรียบเทียบไฟล์

ไฟล์ 1 –nt ไฟล์ 2 จะเป็นจริงเมื่อไฟล์ 1 ใหม่กว่าไฟล์ 2

ไฟล์ 1 –ot ไฟล์ 2 จะเป็นจริงเมื่อไฟล์ 1 เก่ากว่าไฟล์ 2

ไฟล์ 1 –ef ไฟล์ 2 จะเป็นจริงเมื่อไฟล์ 1 มีอีกชื่อเป็นไฟล์ 2

ตัวดำเนินงานเกี่ยวกับสตริง

สตริง 1 = สตริง 2 สตริง 1 เท่ากับสตริง 2

สตริง 1 ! = สตริง 2 สตริง 1 ไม่เท่ากับสตริง 2

–z สตริง ขนาดความยาวของสตริงเป็น 0

–n สตริง ขนาดความยาวของสตริงไม่เท่ากับ 0

ตัวเชื่อมทางตรรกศาสตร์

หลาย ๆ ครั้งการพิจารณานั้นต้องอาศัยหลายเงื่อนไขและหลายเหตุผลประกอบกัน โดยอาจมีการเชื่อมด้วย

ความหมาย "และ" "หรือ" ก็ได้ ในคอร์นเชลล์ได้มีการกำหนดตัวเชื่อมนี้เพื่ออำนวยความสะดวกแก่ผู้ใช้งาน โดยได้มีการกำหนดตัวเชื่อม 3 ชนิดคือ

! (not) จะให้ค่าตรงข้ามกับค่าที่เป็นอยู่

เงื่อนไข 1 –a เงื่อนไข 2 (and) จะให้ค่าเป็นจริงก็ต่อเมื่อเงื่อนไขทั้ง 2 เป็นจริงทั้งคู่

เงื่อนไข 1 –o เงื่อนไข 2 (or) จะให้ค่าเป็นจริงก็ต่อเมื่อเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริง หรือเงื่อนไขทั้งคู่เป็นจริง

คำสั่งที่ใช้ในการตรวจสอบ

คอร์นเชลล์มีเงื่อนไขใช้ในการตรวจสอบมากมายไม่ว่าจะเป็นชนิดของไฟล์สภาวะการอนุญาตของไฟล์ การเปรียบเทียบไฟล์ หรือแม้แต่การเชื่อมเงื่อนไขต่าง ๆ เข้าด้วยกัน ซึ่งไม่ใช้ว่าการตรวจสอบจะมีเฉพาะกับไฟล์เท่านั้น ยังมีการเปรียบเทียบสตริงและค่าทางคณิตศาสตร์อีกด้วย

คอร์นเชลล์ ได้มีคำสั่งในการตรวจสอบเงื่อนไขต่าง ๆ คือ test ซึ่งในคอร์นเชลล์รุ่นหลัง 6 มี.ค. 86 ได้เพิ่มรูปแบบคำสั่งในการตรวจสอบเงื่อนไขโดยสามารถใช้ [[...]] แทน test ได้ด้วย

คำสั่งที่ใช้คำนวณทางคณิตศาสตร์

เชลล์โดยทั่วไปแล้วจะไม่มีกลไกในการคำนวณค่าทางคณิตศาสตร์ ซึ่งเป็นส่วนสำคัญมากในการเขียนโปรแกรม หากขาดส่วนนี้ไปแล้วพูดได้เลยว่าประสิทธิภาพจะต่ำลงมาก ดังนั้นเชลล์รุ่นใหม่ได้แก้ปัญหาโดยมีคำสั่งพิเศษ ซึ่งใช้ในการคำนวณการคณิตศาสตร์โดยเฉพาะคือ expr ดังนี้

```
$ count = 10
$count = $count + 1
$ echo = count
0 + 1
$ expr 1 + 2
3
$ count = 12
$ expr $count + 1
13
$
```

if-then

if-then เป็นรูปแบบที่ใช้ในการตรวจสอบเงื่อนไขที่ง่ายที่สุด โดยมีรูปแบบของคำสั่งดังนี้

```
if (เงื่อนไข)
then คำสั่งที่ต้องทำ เมื่อเงื่อนไขถูกต้อง
     .....
fi
```

เมื่อการตรวจสอบเงื่อนไขถูกต้องเชลล์สคริปต์จะทำตามคำสั่งหลัง then จนกระทั่งครบทุกคำสั่ง โดยจะสิ้นสุดเมื่อพบ "fi" ในกรณีที่การตรวจสอบไม่ตรงกับเงื่อนไขหรือให้ค่าออกมาเป็น 1 ก็จะดำเนินการต่อไปตามคำสั่งหลัง "fi" ดังตัวอย่างรูปที่ 2

if-then else

if-then else เป็นอีกรูปแบบหนึ่งของ if ซึ่งจะมีการตรวจสอบเงื่อนไขที่อยู่หลัง if หากเงื่อนไขเป็นสตริงจะกระทำคำสั่งหลัง then ส่วนในกรณีที่เป็นเท็จก็จะทำคำสั่งหลัง else แทน โดยมีรูปแบบดังนี้

```
if    (เงื่อนไข)
then  คำสั่งที่จะกระทำเมื่อเงื่อนไขเป็นจริง
else  คำสั่งที่จะกระทำเมื่อเงื่อนไขเป็นเท็จ
fi
```

ตัวอย่างดังรูปที่ 3

ลักษณะของ if-then else สามารถกำหนดคำสั่งต่าง ๆ ได้ในตามจำนวนที่ต้องการ และโครงสร้างอีกรูปแบบหนึ่งที่ใช้ร่วมกับโครงสร้าง if-then คือ elif (else if) ซึ่งจะช่วยในการตรวจสอบเงื่อนไขและควบคุมทิศทางการดำเนินงาน เช่นตัวอย่างรูปที่ 4

```
echo "What is your name?\c"
read name
if test "$name" = Somchai
then echo "Your name is very poppular!"
fi
echo "hello $name"
```

รูปที่ 2

```
echo " What is your name ? \c"
read name
if test "$name" = Somchai
then echo ------------------------------
     echo "  Your name is very popular!"
     echo ------------------------------
else echo "hello $name"
fi
echo
echo "This line is out of structure"
```

รูปที่ 3

```
echo "What is your name ? \c "
 read name
 if test "$name" = Somchai
 then echo ------------------------------
      echo "  Your name is very popular!"
      echo ------------------------------
 elif test "$name" = Preeda
 then echo ------------------------------
      echo "  Your name is very basic! "
      echo ------------------------------
 else echo -------------
      echo "  Hello $name"
      echo -------------
  fi
 echo
 echo "This line is out of structure"
```

รูปที่ 4 การตรวจสอบเงื่อนไขและควบคุมทิศทาง

```
echo "What is your name ? \c "
read name
if test "$name" = Somchai
then echo ------------------------------
     echo "  Your name is very popular!"
     echo ------------------------------
else if test "$name" = Preeda
     then echo ------------------------------
          echo "  Your name is very basic! "
          echo ------------------------------
     else if test "$name" = moobin
          then echo ------------------------------
               echo "  Your name is very pretty."
               echo ------------------------------
          fi
     fi
fi
echo " Hello $name"
echo " This end of Program"
```

รูปที่ 5 โครงสร้างของ if-then

```
x=1
  while test $x -le 10
  do
       echo "  Somchai $x"
       x=`expr $x + 1`
  done
echo  ------------------------------------
echo "  This line is out of while loop."
echo  ------------------------------------
```

รูปที่ 6

จากตัวอย่างจะเห็นได้ว่ารูปแบบโครงสร้างของเชลล์สคริปต์มีลักษณะคล้ายกับโครงสร้างของภาษาชั้นสูง เช่น ภาษาซี เลยทีเดียว และในโครงสร้างของ if-then นั้นยังสามารถจัดเป็นโครงสร้างซ้อน ๆ กันในลักษณะของตาข่ายได้ด้วย ดังตัวอย่างรูปที่ 5

จากตัวอย่างต่าง ๆ จะเห็นว่าจะมีการใช้ test command ทุกตัวอย่าง มิได้หมายความว่าในโครงสร้าง if-then ต้องใช้คำสั่ง test เสมอไป เพราะในคำสั่งต่าง ๆ จะมีการรีเทอร์นค่าออกมาอยู่แล้วว่าคำสั่งนั้นถูกต้องหรือไม่ โดยจะให้ค่าเป็น 0 เมื่อคำสั่งนั้นถูกต้อง และจะให้ค่าเป็นอื่น ๆ เมื่อคำสั่งนั้นเป็นเท็จดังตัวอย่างนี้

```
if cat testfile
then echo "this true condition or command"
else echo "This failed condition or command"
fi
```

จากตัวอย่าง ถ้าสามารถ cat testfile ออกมาได้ ก็จะมีการรีเทอร์นค่าเป็น 0 แล้วทำตามคำสั่งหลัง then หากไม่สามารถ cat testfile ออกมาได้ค่าที่รีเทอร์นออกมาไม่ใช่ 0 ก็จะทำตามคำสั่งหลัง else

While

While เป็นโครงสร้างที่เป็นคำสั่งลูป (loop) คือจะเป็นคำสั่งที่ใช้ควบคุมการทำงานที่มีรูปแบบของชุดคำสั่งที่ซ้ำ ๆ กันหลาย ๆ ครั้ง โดยมีรูปแบบโครงสร้างดังนี้

```
While เงื่อนไข
do    คำสั่งที่กระทำเมื่อเงื่อนไขนั้นเป็นจริง
done
```

โดยให้ดูตัวอย่างในรูปที่ 6

การทำงานจะสิ้นสุดเมื่อการตรวจสอบแล้วพบว่าเงื่อนไขเป็นเท็จหรือค่า

รูที่ 7

```
for name in aew leang Somchai Preeda Marsha
do
   echo " Hello, $name."
done
  echo ------------------------
  echo  "  The Moobin Gangs."
  echo ------------------------
```

exit code ไม่ใช่ 0 แต่บางครั้งการกำหนดคำผิดพลาดอาจทำให้เกิด loop ไม่มีที่สิ้นสุดได้ จะออกจากการทำงานในโครงสร้างได้โดยการอินเตอร์รัพต์โปรแกรม เช่น kill command ดังตัวอย่างดังนี้

```
while true
do
echo "Somchai"
done
echo "You cant't fine this line"
```

จะเห็นได้ว่าในตัวอย่างข้างบนไม่มีทางที่จะหลุดจากโครงสร้างได้เลย จนกว่าจะมีการอินเตอร์รัพต์โปรแกรม

For

for เป็นโครงสร้างคำสั่งที่ใช้ในการควบคุมการทำงานของคำสั่งซ้ำ ๆ เดียวกับ while แต่ต่างกัน คือคำสั่ง for จะมีการทำตามคำสั่งตามที่ถูกกำหนดไว้ โดยมีลักษณะโครงสร้างดังนี้

```
for ตัวแปร in บรรดาค่าที่กำหนดไว้
do
      ค่าที่กระทำเมื่อตัวแปรนั้นอยู่ในคำสั่ง
      ที่กำหนดไว้
done
```

ดังตัวอย่างรูปที่ 7 การทำงานในคำสั่ง for นั้นจะสิ้นสุดเมื่อค่าที่กำหนดนั้นหมด

CASE

จากข้างต้นเงื่อนไขที่กล่าวมาจะเป็นลักษณะที่มีทางเลือก 2 ทาง แต่ในความเป็นจริงแล้ว การเลือกนั้นมีได้หลายทางมาก หากใช้เพียงแต่โครงสร้างของ if then หรือ if then else นั้นมาสร้างเป็นโครงสร้างตาข่ายนั้น อาจทำให้เกิดความยุ่งยากขึ้นได้ ดังนั้นจึงได้มีโครงสร้างคำสั่ง case ขึ้นมาเพื่อใช้สำหรับเงื่อนไขที่มีหลายทาง โดยมีรูปดังรูปที่ 8

```
case      ค่าตัวแปร
in
          รูปแบบที่ 1) คำสั่ง ;;
          รูปแบบที่ 2) คำสั่ง ;;
              :
              :
          รูปแบบที่ n) คำสั่ง ;;
esac
```

รูปที่ 8

```
ex
echo "Enter the number"
case "$1"
in
     0) echo zero ;;
     1) echo one ;;
     2) echo two ;;
     3) echo three ;;
     4) echo four ;;
     5) echo five ;;
     6) echo six ;;
     7) echo seven ;;
     8) echo eight ;;
     9) echo nine ;;
     *) echo error ;;
        esac
```

การสร้างบรรยากาศในการทำงาน

โดยปกติแล้วเมื่อเชลล์รับข้อมูลจากคีย์บอร์ดแล้วจะปรากฏเครื่องหมายพรอมต์ (PROMPT) ที่เป็น $ หากเราต้องการเปลี่ยนแปลงเครื่องหมายพรอมต์ให้มีรูปแบบที่ต่างไปก็สามารถแก้ไขได้โดยกำหนดค่าให้กับตัวแปร PS1 ใหม่ได้ดังนี้

```
$ echo "prompt is $PS1"
  prompt is $
$ PS1="> "
> date
  Thu Apr 09 15:58:00 THT 1992
```

ส่วนเครื่องหมายพรอมต์ในเชลล์สคริปต์ จะเก็บอยู่ในตัวแปร PS2 ซึ่งก็สามารถเปลี่ยนแปลงแก้ไขได้ในลักษณะเดียวกับ PS1 ดังนี้

```
> PS2="= = =>"
> for n in 1 2 3 4
  = = => do
  = = => echo $n
  = = => done
  1
  2
  3
  4
```

ยูนิกซ์มีไฟล์พิเศษอยู่ไฟล์หนึ่ง ซึ่งจะเก็บชุดคำสั่งต่าง ๆ ซึ่งจะทำงานทุกครั้ง เมื่อมีการขอใช้งานระบบไฟล์พิเศษนี้คือ .profile ซึ่งคำสั่งส่วนใหญ่จะเป็นคำสั่งที่ใช้บ่อยและจำเป็น เช่น mail การกำหนดรูปแบบของพรอมต์ การเซต path และการกำหนดประเภทของเทอร์มินัล

```
TERM = ansi
HOME = /u/snp
LOGNAME = snp
MAIL = /usr/mail/snp
MAILMSG = YOU HAVE MAIL
PATH = /bin:/usr/bin:/u/snp/-bin
PS1 = $
PS2 = >
SHELL = /bin/ksh
export TERM HOME LOGNAME MAIL MAILMSG PATH PS1 PS2 SHELL
```

สรุป

เชลล์เป็นส่วนหนึ่งที่ทำให้ระบบยูนิกซ์มีประสิทธิภาพสูงขึ้นเป็นอย่างมาก โดยมีเชลล์ที่กล่าวในที่นี้เป็นเพียงส่วนย่อยเท่านั้น เพื่อให้ทราบถึงการทำงานและขีดความสามารถของเชลล์รุ่นหลังอย่างคร่าว ๆ โดยในที่นี้ยังไม่ได้กล่าวถึงการสร้างฟังก์ชันในเชลล์สคริปต์และอื่น ๆ อีกมาก หากท่านผู้อ่านสนใจจะสามารถอ่านเพิ่มเติมได้จากหนังสือ THE KORNSHELL Command and Programming Language เขียนโดย Morris I. Bolsky และ David G.Korn ผู้ที่สร้าง KORN SHELL สำหรับเรื่องที่น่าสนใจอื่น ๆ ในระบบยูนิกซ์ จะนำมาเสนอในคราวต่อไป สวัสดีครับ

SHOW PARTNER F/X เป็นโปรแกรมที่ช่วยในการนำเสนอภาพพร้อมคำบรรยาย โดยอาจเป็นภาพนิ่งหรือภาพเคลื่อนไหวที่มีชีวิตชีวาก็ได้ หนังสือเล่มนี้เขียนขึ้นเพื่อเป็นคู่มือในการใช้โปรแกรม SHOW PARTNER F/X สำหรับผู้ใช้คอมพิวเตอร์และผู้สนใจทั่ว ๆ ไป เนื้อหาภายในเล่มกล่าวถึงพื้นฐานการใช้โปรแกรม SHOW PARTNER F/X เริ่มตั้งแต่ผู้ที่ไม่เคยใช้โปรแกรมนี้ (แต่พอมีความรู้พื้นฐานทางคอมพิวเตอร์บ้าง) จนถึงขั้นที่ผู้อ่านสามารถนำโปรแกรมนี้ไปใช้งานได้

คู่มือการใช้โปรแกรม
SHOW PARTNER F/X

โดย บริษัท ไบต์บิลด์-โลเบิร์ส จำกัด เรียบเรียงโดย พล.ท.ไพบูลย์ เอมพันธุ์

ราคาเล่มละ 155 บาท

หนังสือดี ซีเอ็ด

หาซื้อตามร้านหนังสือทั่วไปหรือสั่งซื้อโดยตรงได้ที่ บริษัท ซีเอ็ดยูเคชั่น จำกัด 800/43-45 ซอยตระกูลสุข ถนนอโศก-ดินแดง ห้วยขวาง กรุงเทพฯ 10400 โดยส่งธนาณัติสั่งจ่าย ปท. ดินแดง ในนามบริษัทฯ หรือเช็คส่วนตัวของธนาคารในเขตกรุงเทพฯ พร้อมค่าจัดส่ง 8% พิเศษ สมาชิกวารสารของซีเอ็ด สั่งซื้อโดยตรงที่บริษัทฯ ได้ในราคาลด 15% เพิ่มค่าลงทะเบียน 4 บาท ค่าจัดส่งฟรี โดยระบุหมายเลขสมาชิก